21.   The computer program product of claim 18, wherein said program code in said computer readable medium for temporarily restricting the execution of said mutators includes a first program code for restricting said mutators from creating any new data objects, and a second program code for temporarily suspending execution of said mutators.

22.   The computer program product of claim 18, wherein said program code in said computer readable medium for completing the garbage collection cycle further includes program code for tracing one or ore pointers stored in each of said mutator thread states.

β1
Cm't

23.   The computer program product of claim 18, wherein said program code in said computer readable medium for completing the garbage collection cycle further includes program code for tracing one or more pointers stored in said objects, for each one of said objects that is in turn pointed to by at least one pointer stored in at least one of said mutator thread states.

24.   The computer program product of claim 18, wherein said program code in said computer readable medium for completing the garbage collection cycle is performed without copying any object in said plurality of data objects.

25.   The computer program product of claim 18, wherein said program code in said computer readable medium for completing the garbage collection cycle further includes program code for executing said mutators subject to a write barrier which is invoked when said mutators modify pointers within said plurality of data objects but not when said mutators modify pointers

within said thread states.

26.   A method for invoking a write barrier during real-time computer garbage collection, for use with a plurality of data objects and with one or more mutator programs, each one of said mutators having a corresponding thread and each one of said mutator threads having a corresponding thread state separate from said plurality of data objects, said method comprising the steps of:

temporarily restricting execution of said mutators during a garbage collection cycle, while processing the corresponding thread state for each one of said mutators;

permitting each one of said mutators to resume unrestricted execution, as soon as said mutator's own corresponding thread state has been processed;

identifying each one of said objects that is currently accessible to at least one of said mutators; and

executing said mutators subject to a write barrier which is invoked when said mutators modify pointers within said plurality of data objects but not when said mutators modify pointers within said thread states.

27. The method of claim 26, wherein said step of temporarily restricting the execution of said mutators includes a first step of restricting said mutators from creating any new data objects, and a second step of temporarily suspending execution of said mutators.

28. The method of claim 26, which further comprises:

tracing one or more pointers stored in said objects, for each one of said objects that is in

turn pointed to by at least one pointer stored in at least one of said mutator thread states.

29. An apparatus for performing real-time computer garbage collection, for use with a plurality of data objects and with one or more mutator programs, each one of said mutators having a corresponding thread and each one of said mutator threads having a corresponding thread state separate from said plurality of data objects, said apparatus comprising:

a garbage collector for processing the corresponding thread states of said mutators at the beginning of a garbage collection cycle, and for identifying each one of said data objects that is accessible to at least one of said mutators during said cycle;

one or more processors for executing said garbage collector and said mutators; and

a scheduler coupled to said processors for scheduling execution of said garbage collector and said mutators on said processors, said scheduler being operative to temporarily restrict execution of said mutators while the corresponding thread state is being processed for each one of said mutators, and to permit each one of said mutators to resume unrestricted execution as soon as said mutator's own corresponding thread state has been processed.

30. The apparatus of claim 29, wherein said garbage collector further includes a processor for flipping said one or more objects from a first label representative of accessible status to a second label representative of undetermined status.

31. The apparatus of claim 29, wherein said garbage collector further includes a storage for saving a list of said mutator threads.

32. The apparatus of claim 29, wherein said garbage collector includes a processor for tracing one or more pointers stored in each of said mutator thread states.

33. The apparatus of claim 29, wherein said garbage collector includes a processor for tracing one or more pointers stored in said data objects, for each one of said objects that is in turn pointed to by at least one pointer stored in at least one of said mutator thread states.

34. The apparatus of claim 29, wherein said garbage collector does not copy said data objects.

35. The apparatus of claim 29, wherein said one or more processors are implemented with stock hardware.

36. The apparatus of claim 29, wherein said scheduler is operative to temporarily restrict said mutators from creating any new data objects, and is further operative to suspend execution of said mutators.

37. The apparatus of claim 29, further including a processor for executing said mutators subject to a write barrier which is invoked when said mutators modify pointers within said plurality of data objects but not when said mutators modify pointers within said thread states.

REMARKS

No new matter is presented by the foregoing amendments. Entry and an early allowance is respectfully requested.